Modeling Basic Maneuvering with the Predator UAV

Kevin Gluck
Air Force Research Laboratory
kevin.gluck@williams.af.mil

Jerry Ball
L-3 Communications
jerry.ball@williams.af.mil

Mike Krusmark
L-3 Communications
michael.krusmark@williams.af.mil

Background

Air Force combat units find it increasingly challenging to implement effective training programs. Range restrictions, budget cuts, aging or inoperable equipment, and the current ops tempo make it difficult to maintain desired levels of mission ready personnel. To address these training challenges, the Air Force Research Laboratory's Warfighter Training Research Division has pioneered a new training capability called Distributed Mission Training (DMT). DMT is a networked system of live, virtual, and/or constructive entities that fight with or against each other in a synthetic battlespace. The ultimate vision for DMT is nothing less than anytime, anywhere, individually-tailored training for every warfighter in the combat Air Force.

As with any grand vision, there are innumerable hurdles yet to overcome. Many of these hurdles involve the science of human behavior representation in modeling and simulation, because the desired end state is a realistic and effective training capability in which the trainee is the only human required. In this vision, all necessary friendly and opposing forces, as well as the trainee's instructor, can be represented synthetically and called upon to participate in training exercises, if need be. It will be the ultimate intelligent tutoring system. It is a long way off.

The Performance and Learning Models Research Program

In response to these challenges, AFRL's Warfighter Training Research Division has established the Performance and Learning Models Research Program at the Mesa Research Site, for conducting basic and applied research in human behavior representation. The vision motivating creation of this lab is a revolution in training – more effective and more efficient training leading to a higher level of readiness, made possible through basic and applied cognitive science. We intend to achieve that vision through a dual-emphasis approach involving *empirical research* and *computational process modeling*.

The Performance and Learning Models Lab recently received funding from the Air Force Office of Scientific Research (AFOSR) for a basic research program focusing on visuospatial working memory (VSWM) and the Uninhabited Air Vehicle (UAV) Operator. VSWM is the set of cognitive processes that allow people to visualize the relative positions of things. It can be experimentally distinguished from working memory for verbal materials. Visuospatial working memory is said to be involved in virtually all spatial problem solving and may be particularly crucial for operators of UAV's. To test this hypothesis, we are developing computational process models of selected components of UAV operation, specifically tasks having to do with the piloting of the aircraft by an Air Vehicle Operator (AVO).

UAV Operator Model

The longer-term modeling goal is a UAV Operator model capable of skillfully executing a reconnaissance mission with the Predator UAV STE. This complex skill is composed of a variety of simpler skills, such as the skills required for basic maneuvering of the UAV. Conveniently, basic maneuvering is one of the tasks included in the UAV STE, and therefore a model of basic maneuvering skills has been the shorter-term goal to date.

<u>Unit Tasks in Basic Maneuvering</u>

In other recent ACT-R models of human-system interaction with dynamic tasks, a modeling idiom has emerged involving the representation of what Newell called "unit tasks." These are recurring sub-tasks that make skilled performance in the broader task possible. To further test the generalizability of unit tasks as a model design idiom, we have adopted a unit task representation with the UAV Operator model. In this model, the unit tasks are the major performance indicators in the basic maneuvering segments: heading, speed, and altitude.

In the 1st basic maneuvering task, the Operator's goal is to maintain current heading (0°) and altitude (15,000 ft.) while steadily decreasing speed from 67 knots to 62 knots, over the course of a 1-minute trial. The assumption made in the model is that the Operator's goal of flying the aircraft in such a way as to meet these performance criteria can be satisfied by continuously cycling through the three corresponding unit tasks. Currently, the model selects among these unit tasks randomly. After completing a unit task, the model returns to the initial state and selects another unit task. All of the unit tasks involve the same abstract process. For illustrative purposes, let's assume the model chooses to check its heading. Step 1 in the Heading unit task is that the model retrieves declarative knowledge of the location of the heading indicator on the HUD. ACT-R's visual module then locates a screen object at this location, moves visual attention to that location, and encodes the value there. Now the model has encoded the aircraft's current heading. The model can now compare the current heading with the desired heading and determine if any action is required. If so, it executes that action. If not, the model randomly selects another unit task.

Model Development

A key factor for the UAV Operator model development is the existence of a UAV Synthetic Task Environment (STE). The STE system includes 11 separate processes running on 2 separate computers—an Instructor Operator Station (IOS) and a Pilot Station. The STE is coded in C and uses a proprietary datagram socket and shared memory interface for communication among processes. The STE also includes a "heartbeat" mechanism for keeping the separate processes synchronized. The source code for many of the processes is available, but not for the core "heartbeat" process. The reality of working with an existing system meant that the UAV Operator model had to communicate with the STE using datagram sockets and/or shared memory and that it was the responsibility of the model to synch up with the STE and not vice versa. As it turns out, the UAV Operator model with its graphic display of a mockup of the Heads Up Display and graphs of the control inputs is actually slower than the STE—despite claims that ACT-R models can run faster than real time. Thus, it was necessary to speed up the UAV Operator model to keep it in sync with the STE. Communication with the STE was effected by developing a separate C process running on the same machine as the UAV Operator model. This C process communicates with the STE using datagram sockets and communicates with the UAV Operator model using shared memory. The control inputs process of the STE had to be modified to support control inputs from the UAV Operator model instead of the actual stick, throttle and rudder. The windows processing loop of the control inputs process was modified to trap a datagram socket arrival event (containing the control inputs) and process it. Processing of the datagram socket arrival event is not part of the basic communication mechanism of the STE which uses a "heartbeat" event and datagram polling to retrieve datagram sockets from other processes.

Initial development occurred using ACT-R 4.0 and Allegro CL 5.0.1. The Allegro CL sockets library does not support non-blocking sockets making using of a datagram socket polling mechanism problematic and necessitating introduction of a separate C process on the UAV Operator model station to perform this function. The subsequent availability of ACT-R 5.0 made it possible to take advantage of the visual component of RPM and a mock Heads Up Display (HUD) corresponding to the HUD of the UAV STE was developed. It was not, however, possible to take advantage of the motor component of RPM since there is no support for Hands On Throttle and Stick (HOTAS) in that component. Instead, a delay factor of 200 msec was added to productions that effect stick and throttle movements.

A key issue for the use of RPM is how to avoid having to reimplement the STE in the UAV Operator model in order to take advantage of RPM functionality.

We briefly looked at a Java implementation of ACT-R since the free availability of Java software made this attractive. However, the Java version was not yet ready for serious model development and varied from ACT-R 5.0 in significant ways (e.g. no RPM support).

The selection of the "Unit Task" pattern guided model development. The "enable randomness" parameter of ACT-R was added in order to support the random selection of units tasks during model runs. This is an initial approach that will need to be refined.

Three unit tasks were identified: airspeed, altitude and heading. Two of these tasks—airspeed and altitude—involve significant interaction, with control inputs from stick and throttle affecting both airspeed and altitude simultaneously and in fairly complex ways. The modeling of these two tasks attempts to account for the interactions. Heading interacts only mildly with airspeed and altitude under normal conditions and is modeled independently of the other tasks.

Interviews with a local aviation expert and a UAV Operator guided development. A table reflecting the selection of stick and throttle inputs under varying altitude, airspeed and vertical airspeed (VSI) conditions was created and folded into the development of procedures for the altitude and airspeed tasks. Equations were developed to determine the size of stick and throttle adjustments and were based on the size of the deviation of the current airspeed/altitude/VSI from the desired values.

Very recently we have become aware that the performance of the model varies between different instances of the 4 available STE's. That is, the calibration of the control inputs to the STE's differs, resulting in different behaviors. The UAV Operator model is unable to adjust to these calibration differences.

Model Assessment

To date, our primary concerns have been (1) creating a baseline ACT-R model and (2) establishing reliable communication between the simulation and the model. Only recently have we begun to assess the extent to which the model predicts the behaviors of human Air Vehicle Operators (AVO's). The data we have available for such an assessment include: 1) active-duty AVO's who participated in an earlier STE validation study, and (2) verbal protocols and eye movement data from a retired AVO who visited our lab. The latter data remain mostly un-analyzed, so here we focus on the former.

The data available from active-duty AVO's take two forms: flight control settings (throttle and stick) measured continuously throughout each trial, and performance outcome measures showing average altitude, airspeed, and heading deviations at the end of each trial. Comparisons of model predictions to active-duty AVO data show the model's control inputs to be a good characterization of the control actions taken by the AVO's. The magnitude and the direction of the inputs are a good fit. When we examine outcome performance data, however, the results are not as positive. Here we are comparing the model's average deviation from the ideal flight path with averages by the human AVO's. It is clear that, on all three measures (altitude, airspeed, and heading) the model's performance is substantially worse than the AVO's.

Clearly, the model is not yet an accurate representation of the knowledge and skill of an active-duty AVO, and this raises the issue of what known shortcomings are still present in the model. There are many, but we will highlight three here. One is that the magnitude of the control inputs used by the model is computed with embedded lisp equations. Although this does not account for the model's poor performance, it is a shortcoming because the equations are not learnable within the architecture, which should be a hallmark of all knowledge representation in ACT-R models. Another shortcoming is that the unit tasks are always selected randomly. A more valid representation would involve dynamic priority setting for attention to indicators related to "problem areas" during the trial. For instance, if the model notes that its heading has drifted considerably, it should increase its focus on heading until the problem is resolved. A third shortcoming is the reticle and horizon line are not yet available to the model. This is a serious and rather obvious ommission, but a capability that is challenging to implement. We will have to overcome this challenge if we are ever to claim to have a model that uses knowledge and skills similar to those of a real AVO.

Next Steps

Improvements to the model will focus first on addressing the known shortcomings listed above, as well as others we have identified. In parallel with those improvements, we will be extending the model so that it is able to fly the remaining six basic maneuvering segments. After that, the goal is to further extend the model so that it is able to fly reconnaissance missions. Ultimately, the model will become a tool for studying the impact of individual differences on mission success, and for exploring ways to represent the learning processes taking place in the context of UAV operation.